Journal of Network and Computer Applications ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks

Jin-Hee Cho [a,*], Ananthram Swami [a], Ing-Ray Chen [b,1]

[a] Computational & Information Sciences Directorate (CISD), U.S. Army Research Laboratory (USARL), 2800 Powder Mill Rd, Adelphi, MD 20783, United States
[b] Department of Computer Science, Virginia Polytechnic State University, 7054 Haycock Road, Falls Church, VA 22043, United States

ARTICLE INFO

Article history:
Received 2 August 2010
Received in revised form
19 January 2011
Accepted 9 March 2011

Keywords:
Mobile ad hoc networks
Trust
Trust metric
Trust management
Quality-of-service trust
Social trust
Stochastic Petri nets

ABSTRACT

We develop and analyze a trust management protocol for mission-driven group communication systems in mobile ad hoc networks using hierarchical modeling techniques based on stochastic Petri nets. Trust among mobile nodes is crucial for team collaborations with new coalition partners without prior interactions for mission-driven group communication systems in battlefield situations. In addition, ensuring a certain level of trust is also critical for successful mission completion. Our work seeks to identify the optimal length of a trust chain among peers in a *trust web* that generates the most accurate trust levels without revealing risk based on a tradeoff between trust availability and path reliability over trust space. We define a trust metric for mission-driven group communication systems in mobile ad hoc networks to properly reflect unique characteristics of trust concepts and demonstrate that an optimal trust chain length exists for generating the most accurate trust levels for trust-based collaboration among peers in mobile ad hoc networks while meeting trust availability and path reliability requirements.

Published by Elsevier Ltd.

1. Introduction

Mobile ad hoc networks (MANETs) are defined as multi-hop wireless networks dynamically formed by mobile nodes, operating without the help of any centralized infrastructure (Tardiff and Gowens, 2001). Group communication systems (GCSs) in MANETs, such as in military battlefields or first responder scenarios, require teamwork and collaboration to achieve a mission that depends on the trust relationships among group members (Li et al., 2008). Blaze et al. (1996) first introduced the term *trust management* and identified it as a separate component of security services in networks. Trust management in MANETs also has received considerable attention due to its crucial necessity and diverse applicability. Trust management in MANETs is needed when participating nodes, without any previous interactions, desire to establish a network with an acceptable level of trust relationships among themselves, for example, in building initial trust bootstrapping, or in coalition operation without predefined trust. Trust management is also required in the collection and distribution of evidences to assess or maintain the levels of trust required for successful task completion. In addition, trust management has diverse applicability in many decision making situations requiring collaboration of participating nodes with goals such as secure routing, key management, intrusion detection, authentication, and access control (Capkun et al., 2003; Li and Singhal, 2007; Theodorakopoulos and Baras, 2004).

In addition to the challenges faced in the design of stationary wired networks, security protocol designers for GCSs in MANETs face technical challenges due to the unique characteristics of MANETs such as resource-constraints (e.g., bandwidth, memory, energy, computational power), openness to eavesdropping, high security threats or vulnerabilities, inherent unreliability of communications over a wireless medium, and rapid changes in topologies or memberships due to node mobility or failure (Tardiff and Gowens, 2001). Security protocols designed for military MANETs must also take into account unique scenarios not typically encountered in civilian applications, such as hostile environments, proneness of nodes to capture and subversion, heterogeneity of nodes and of interacting networks, support dynamic communities of interest and coalition operations, ability to reconfigure rapidly to cope with dynamics, while at the same time ensuring that network control and operations are not complex (Plesse et al., 2004).

In order to construct trustable collaborative environments based on unique characteristics of MANETs, many researchers (Abdul-Rahman and Hailes, 1997; Bhargava et al., 2004; Blaze

* Corresponding author. Tel.: +1 301 394 0492.
E-mail addresses: jinhee.cho@us.army.mil, jinheechogwb@yahoo.com (J.-H. Cho), ananthram.swami@us.army.mil (A. Swami), irchen@vt.edu (I.-R. Chen).
[1] Tel.: +1 703 538 8376.

1084-8045/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.jnca.2011.03.016

et al., 1996; Capra, 2004; Li and Singhal, 2007; Li et al., 2008; Mahmoud, 2007; Theodorakopoulos and Baras, 2004; Yang et al., 2007) have adopted trust concepts to evaluate the relationships among group members. In the social sciences, trust is defined as the degree of a subjective belief about the behaviors of a particular entity (Cook, 2003).

In developing trust management systems for MANETs, researchers have heavily focused on developing secure ad hoc routing protocols based on trust or reputation with the aim of isolating malicious or selfish nodes for improving system metrics (e.g., end-to-end packet delivery ratio) (Balakirshnan and Varadharajan, 2005; Theodorakopoulos and Baras, 2004; Yang et al., 2007). However, no prior work exists on trust management in MANETs that properly accounts for dynamically changing environments including topology changes, membership changes, energy depletion, nodes' heterogeneity (e.g., different energy levels of nodes), selfishness, healthiness or honesty (i.e., not compromised), and frequency of interactions over time.

Our work takes into account the dynamically changing conditions in MANET environments. Our aim is to design and evaluate a trust management protocol for mission-driven GCSs in MANETs. Evidences for trust may be gathered via direct interactions or indirect interactions reported by nodes over multiple hops. Using longer trust chain length potentially increases the accuracy of estimated trust levels; but there is a tradeoff with latency, overhead and risk. The value of the reported evidence decays over time and space. We seek to characterize the tradeoff between trust accuracy and path reliability in terms of the length of a trust chain. Our trust management protocol takes into account the presence of selfish and malicious (i.e., inside and outside attackers) nodes when participating nodes are assumed to have no prior interactions.

Trust availability is measured by the probability that a trustor (or evaluator) node can find a target node within a specified number of hops to obtain its trust level. Trust decays as the trust chain grows, and is measured by reliability of a route that the trust information of a remote node is passed by intermediate nodes, called *path reliability*.

The contributions of this work are as follows. First, we develop and evaluate a *trust metric* that reflects unique characteristics of GCSs in MANET environments such as subjectivity, asymmetry, transitivity, dynamicity, and context-dependency. Second, we develop a *mathematical model* based on *stochastic Petri net (SPN)* techniques to evaluate design tradeoffs. In particular, we develop a hierarchical modeling technique to avoid state explosion problems and to efficiently calculate the trust levels of a large number of nodes. Third, we incorporate concept of social networks inspired by theories from social sciences to model trust-based GCSs in MANETs. We model the social behavior of a node in the social network by its *social trust* viewed by other peers in the network. Fourth, we identify the optimal trust chain length that would generate the most accurate trust levels of peers on the trust chain based on the tradeoff between trust availability and path reliability as the length of the trust chain grows.

The rest of this paper is organized as follows. Section 2 surveys related work in the literature. Section 3 discusses trust concepts in MANETs, provides backgrounds on social networks, and describes our proposed trust management protocol. Section 4 describes the system model and assumptions, the main design features, defines the trust metric, and describes the attack model. Section 5 develops the SPN performance model and describes how the SPN model can be used to evaluate system behaviors under the proposed trust management protocol. Section 6 gives numerical results obtained through the evaluation of our performance model and provides their physical interpretations. Finally, Section 7 concludes our paper and suggests future work.

2. Related work

Evidence-based trust management and *monitoring-based trust management* (Li and Singhal, 2007) are two popularly accepted methodologies to evaluate trust values of a node; see, e.g., Balakirshnan and Varadharajan (2005), Blaze et al. (1996), Theodorakopoulos and Baras (2004), and Yan et al. (2003). Evidence-based trust management considers anything that proves the trust relationships among nodes including public key, address, identity, or any evidence that any node can generate for itself or other nodes through a challenge and response process. Monitoring-based trust management rates the trust level of each participating node based on direct as well as indirect information. Direct information includes observations of a node's behaviors such as packet dropping and flooding. Indirect information includes reputation ratings forwarded by other nodes.

Existing trust management or trust-based network protocols in MANETs are developed with diverse purposes such as secure routing, access control, intrusion detection, key management, and authentication. For developing collaborative secure routing protocols in MANETs, many papers seek to detect misbehaving nodes including both selfish and malicious nodes (Ghosh et al., 2005; Li et al., 2008; Mundinger and Boudec, 2008; Nekkanti and Lee, 2004; Pirzada and McDonald, 2004; Soltanali et al., 2007; Sun et al., 2006). Most of these *secure routing* protocols are proposed based on characteristics of trust (e.g., asymmetry, transitivity, subjectivity, dynamicity, and context-dependency) by considering either selfish nodes or malicious nodes or both. However, no prior work comprehensively addresses the unique characteristics of trust in MANETs.

Trust management is also used to ensure *authentication* (Ghosh et al., 2005; Pirzada and McDonald, 2004). The weighted transitivity property of trust was used to extend the trust space based on the concept of a trust chain. However, this work only uses direct information to evaluate trust. Further, they did not seek to optimize the length of the trust chain.

Intrusion detection is also targeted as a goal of trust management schemes in MANETs (Albers et al., 2002; Ahmed et al., 2006). Albers et al. (2002) proposed a trust-based Local Intrusion Detection System (LIDS). Ahmed et al. (2006) leveraged intrusion detection mechanisms in order to evaluate the trust levels of participating nodes. However, their work considered direct observations only. Access control is a popular application of trust management in MANETs (Adams and Davis, 2005; Gray et al., 2002; Luo et al., 2004). Gray et al. (2002) proposed a trust-based admission control based on local and global policies to measure trust level per session in MANETs, but does not consider node dynamics or failure. Luo et al. (2004) proposed a trust-based control system based on threshold cryptography. Adams and Davis (2005) developed an access control mechanism based on both direct observations and reputation to detect misbehaving nodes.

Trust has also been applied in developing *key management* protocols in MANETs (Adams and Davis, 2005; Hadjichristofi et al., 2005; Li et al., 2006; Virenda et al., 2005). Virendra et al. (2005) proposed a trust-based security architecture for secure distributed control in MANETs. However, no analytical results were given to prove the effectiveness of their protocol. Hadjichristofi et al. (2005) presented a key management protocol providing redundancy and robustness of Security Association (SA) establishment between pairs of nodes in MANETs based on a hierarchical trust Public Key Infrastructure (PKI) model where nodes can dynamically take management roles. However, trust relationships are derived solely from certificate chains. Adams and Davis (2005) proposed a node-centric reputation management scheme that considers feedback of a node's behavior with a reputation index. The reputation index is used to weigh a

Please cite this article as: Cho J-H, et al. Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks. J Network Comput Appl (2011), doi:10.1016/j.jnca.2011.03.016

recommender's reputation when evaluating feedback provided by the recommender on the trustworthiness of another node. Li et al. (2006) demonstrated a public key management protocol for providing authentication. Yan et al. (2003) proposed a trust-based solution to provide effective security decisions on data protection, secure routing, and other network activities. Jiang and Baras (2004) proposed a trust distribution scheme, called Ant-Based trust Evidence Distribution (ABED), based on the *swarm intelligence paradigm*. Baras and Jiang (2004) further addressed distributed trust computation and establishment using random graph theory. Theodorakopoulos and Baras (2004) proposed a trust evidence evaluation scheme for MANETs based on the theory of *Seminrings*. The evaluation process is modeled as a path problem in a directed graph where nodes indicate entities and edges represent trust relations. Recently Buckerche and Ren (2008) proposed a distributed reputation evaluation prototype. Moloney and Weber (2005) presented a trust-based security system that generates appropriate trust levels based on the consideration of the unique characteristics of MANETs as well as context-awareness.

Our work differs from the existing work cited above in that we consider a comprehensive set of trust properties in MANETs and we incorporate the trust concept into social networks in developing our trust management protocol.

Our research has its root in quantitative modeling (Ciardo et al., 1994/1999; Sahner et al., 1996). We use an SPN as our mathematical model for performance analysis. An SPN model is essentially a concise representation of a *Markov* or *semi-Markov* model, capable of accommodating a large number of states. It can also accommodate general time distributions other than commonly used exponential time distributions for modeling system events. In the literature on trust management for MANETs, little has been done using quantitative modeling techniques (Li, 2008; Moe et al., 2008; Mundinger and Boudec, 2008). Mundinger and Boudec (2008) used a stochastic process model. Li (2008) modeled *opinion* to represent trust among nodes using *subjective logic*. Moe et al. (2008) described the trustworthiness of a node using the state probability of the node in a *hidden Markov model (HMM)*. Different from our prior work (Cho et al., 2009), this work focuses on identifying the optimal trust chain length for generating the most accurate trust levels evaluated by the new trust metric which embeds the tradeoff between trust availability (i.e., extended trust space) and path reliability (i.e., trust decays over space).

3. Preliminaries

3.1. Properties of trust in MANETs

Due to the unique characteristics of MANETs and the inherent unreliability of the wireless medium, trust management for MANETs should encompass the following trust concepts. First, trust is *dynamic*, not static. Trust in MANETs should be established based on spatially and temporally local information, which may be incomplete or inaccurate due to network dynamics (Capra, 2004). Second, trust is *subjective* (Abdul-Rahman and Hailes, 1997). In MANETs, different nodes may assess different trust levels with respect to a target node due to different experiences with the node. Third, trust is *not necessarily completely transitive* (Sun et al., 2006). For example, if *A* trusts *B*, and *B* trusts *C*, but *A* does not necessarily trust *C*. Fourth, trust is *asymmetric* and not necessarily reciprocal (Adams and Davis, 2005). When *A* trusts *B* with a trust level *a*, it does not mean *B* trusts *A* with the same trust level *a* or may mean B may even distrust *A*. Finally, trust is *context-dependent* (Sun et al., 2006). For example, *A* may trust *B* as a wine expert but not as a car fixer. For more details of trust concepts and properties in MANETs, readers are referred to (Cho et al., 2010).

3.2. The proposed trust management for MANETs

We assume that there is no predefined trust in the initial network deployment. Without previous interactions, the initial bootstrapping will establish a shallow level of trust based only on indirect information (e.g., reputation from historically collected data or recommendation by third parties) and authentication by a challenge/response process (e.g., public key authentication). Over time, nodes will establish a stronger trust level with more confidence among group members based on direct interactions.

Trust is subjective and asymmetric as we described above. Each node has a different capability (e.g., energy, number of 1-hop neighbors, cooperativeness, and honesty) and it may use its own capabilities as criteria to evaluate other nodes. In organizational management (Schoorman et al., 2007), a supervisor tends to trust an employee less than the employee trusts the supervisor. Likewise, in this work, a node with high capability may evaluate other nodes' trust levels more strictly than a node with low capability.

Our protocol allows each node to evaluate the overall trust of other nodes as well as to be evaluated by other nodes based on two factors, social trust and QoS (quality-of-service) trust. *Social trust* may include friendship, honesty, privacy, and social reputation or recommendation derived from direct or indirect information for "sociable" purpose. On the other hand, *QoS trust* may embrace competence (e.g., computational power, radio range, or energy), dependability, successful experience, cooperativeness (unselfishness), and task-oriented reputation or recommendation computed from direct or indirect information for a mission execution purpose.

Trust is context-dependent particularly in dynamic distributed systems such as MANETs. That is, the combination of social trust and QoS trust will contribute to generating the overall trust metric; the weight ratio will vary with the degree of difficulty or risk of the assigned mission, and may dynamically react to operational and environmental network conditions.

Our composite trust metric takes network dynamics into account. Trust decays over time without further updates or interactions between entities. Node mobility also hinders continuous interactions with other group members, lowering the chances of evaluation of each other in the group. This includes cases such as a node moving to other areas causing its disconnection from the current group, leaving a group for tactical reasons, voluntary disconnection (for saving power) or involuntary disconnection (due to physical terrain or low energy). We also assume that individual nodes may behave selfishly in resource-constrained MANET environments.

We adopt the concept of *web of trust* (Capra, 2004; Cook, 2003) in order to expand trust over space based on a weighted transitivity of trust. We obtain a certain degree of trust based on the length of a trust chain. For example, when the length of the trust chain is 4, e.g., *A* trusts *B*, *B* trusts *C*, *C* trusts *D*, and *D* trusts *E*, then *A* may trust *E*. However, the longer the trust chain, the more is the decay in the degree of trust (Capra, 2004). We also use the concepts of referral and functional trust defined by Josang and Pope (2005). Note that recommendations passed by a node is the trust value only based on direct observations or relationships. Particularly, notice that trust information passed by nodes *A*, *B*, *C*, and *D* is referral trust (e.g., they do not have direct relationship with node *D* but know someone who has direct relationship with *D*) while one passed by *D* about node *E* is called functional trust (i.e., node *D* actually has direct relationship with node *E*) (Josang and Pope, 2005).

According to Josang and LoPresti (2004) and Solhaug et al. (2007), if the measured trust (often called the subjective level of trust probability or simply called *trust)* overestimates *trustworthiness,* the objective level of trust probability, then collaboration may be risky, since the chance of being betrayed by a trustee increases. On the other hand, if the measured trust underestimates trustworthiness (actual trust), a trustor may lose benefit by not collaborating with potentially good partners.

Our proposed trust management system is for a GCS in military MANETs where a symmetric key, called the group key, is used as a secret key for communications between group members (Steiner et al., 1996). Upon a node's disconnection from the group, the system generates and redistributes a new key so that non-member nodes will not be able to access a valid secret group key. However, our proposed approach will still allow each group member to keep the old trust information on non-member nodes so that they can reuse the old trust information for future interactions, properly weighted to reflect the decay over time. This may prevent a newcomer attack attempting to flush out its notorious past trust or reputation.

4. System model

4.1. Protocol design and assumptions

We assume a pure MANET environment, without a centralized trusted entity, where nodes communicate through multi-hops. We assume that mobile devices are carried by dismounted soldiers and consider walking speeds $(0,x]$ m/s for node mobility where x indicates the upper bound of a speed of an entity. Nodes may have different levels of energy and speed, thus reflecting characteristics of a heterogeneous network. Each node periodically beacons its *id* and *location* information which enables neighboring nodes to easily detect node failures and maintain valid group membership in a timely manner. We consider a single group with a single assigned mission. Involuntary disconnections or reconnections caused by network topology changes (e.g., network split or merge due to node mobility or failure) are implicitly considered by a node's join or leave process and the corresponding rekeying cost is considered in calculating energy consumption, as shown in the Appendix of the paper. A node's disconnections or reconnections is incorporated in calculating trust values of a node based on the “closeness” trust component which represents the degree of 1-hop neighbors of each node.

We assume that nodes often behave maliciously or selfishly caused by their inherent nature as well as environmental or operational conditions. That is, other than being affected by their given nature, nodes are also affected by operational conditions. For example, a node is much more likely to be selfish to save its own energy particularly when the energy level is low. Further, a node can be compromised. We relate the energy level of a node with the rate at which the node may be compromised. That is, a node is more likely to be compromised when its energy level is low and vice versa since a node with high energy is more capable of defending itself against attackers by performing more energy-consuming defense mechanisms. Note that the association between a node's status and its behavior is based on the assumption that each node has its own inherent nature to trigger bad behaviors.

We assume that there exists a distributed intrusion detection subsystem (IDS) for detecting insider attacks (compromised nodes). As soon as a node is detected by IDS, the node is no longer alive in the system meaning that trust value of the node will drop suddenly. We do not prescribe specific IDS, but assume that its false positive and false negative probabilities are known. We define the selfish behavior of a node as dropping group communication packets transmitted from other nodes. Thus, even though the node is selfish, it cooperates to perform rekeying and IDS-related operations. We also assume that potential attackers, compromised but not detected by IDS, may disseminate bogus packets.

The energy level of each node is adjusted depending on its status. For simplicity, we only consider energy consumption based on the communication mode: receiving or transmitting, without considering idle listening. For example, if a node becomes selfish, the rate of energy consumption is slowed down and vice versa. If a node becomes compromised but not detected by IDS, the rate of energy consumption would grow since the node may have a chance to perform attacks, thus consuming more energy. We also consider redemption mechanisms for selfish nodes. We use a period of reevaluation for selfish nodes, at the end of which a selfish node can determine whether it will resume normal behavior or continue being selfish depending on its own remaining energy level. In addition, when a node is not a member, it will not consume as much energy as when it is a member. We model group member join and leave operations as most GCSs do. Upon every membership change due to join/leave/eviction, individual rekeying will be performed based on a distributed key agreement protocol.

We assume that a node's trust value is assessed based on direct observations and indirect information passed from recommenders. For indirect information, this work uses recommendations obtained from 1-hop neighbors of a target node. We call intermediate nodes passing the recommendation from the 1-hop neighboring recommenders of the target node *referral recommenders*. The 1-hop neighboring recommenders of the target node are called *functional recommenders*. In our protocol, the functional recommenders selected are the ones that have the highest trust values about the target trustee node. The rationale is twofold. First, with the presence of IDS that evicts bad nodes, there is a high probability that the functional recommenders selected are good nodes. Second, our trust metric reflects the amount of interactions between the trustor and the trustee, so selecting those functional recommenders having high trust values about a target node means that these functional recommenders selected have interacted more with and consequently know well about the target node. However, the referral recommenders are selected randomly in order to avoid dominant correlation of recommendations relayed by nodes that are regarded as highly trustable but actually compromised but undetected. Note that all recommenders provide their recommendations based on their direct observations (i.e., direct trust), not based on indirect information (i.e., indirect trust). Josang and Pope (2005) explained that when derived trust (trust evaluated by both direct and indirect trust) is used for recommendations, trust information from a compromised referral node can be passed to multiple places and may cause serious security vulnerability.

If the number of recommendations received is less than the one required (i.e., # of recommendations $< |S|$ in Eq. (6)), recommendations from all 1-hop neighbors (functional recommenders) can be used. Further, when no trust information is received, then each node simply relies on the trust information collected at a previous trust update.

It is assumed that each node can observe behaviors of its 1-hop neighbors and computes trust component values based on the direct observations using a reputation monitoring mechanism such as *Watchdog* or *Pathrater* (Marti et al., 2000). Each node periodically disseminates this direct trust information of its 1-hop neighbors and its own ID using a group key, via so called status exchange messages. This will enable each node to compute trust values of other nodes considering the original recommendation from the 1-hop neighbors of a target node as well as the reliability of the path over which the trust information is obtained. When a node receives the status exchange message, it can calculate trust based on desired trust availability and required path reliability. *Trust availability* is the probability that a target node exists within n-hop distance from an evaluator's location where n is the length of the trust chain used. Thus, as n increases, trust availability increases. On the other hand, when a target node is found within n hops from the evaluator's location, the reliability of the route (called *path reliability*) taken to pass the trust information from a functional recommender will decrease. We calculate path reliability as the product of referral trust values of all referral recommenders (or relay nodes). The referral trust value is measured by the product of two direct trust components values, honesty and cooperation (not faking or not dropping packets). Thus, as n increases, path reliability decreases. Based on this tradeoff, each node adjusts the length of its trust chain in order to collaborate with more nodes by having desired trust availability while maintaining required path reliability.

4.2. *Trust metric*

We consider a trust metric that spans two aspects of the trust relationship. First, *social trust* (Golbeck, 2009) will be evaluated to account for social relationships. We consider honesty and closeness for social trust where honesty refers to the degree of being uncompromised and closeness is measured by how many 1-hop neighbors a node has in the social network. Second, *QoS trust* accounts for the capability of a node to complete a given mission. We consider the energy level and degree of cooperation (or unselfishness) to estimate the QoS trust level of a node. A node's trust value changes dynamically to account for trust decay over time due to node mobility or failure, as the trust chain becomes longer, as the node's energy level changes, as a node becomes compromised or not, and as the node becomes selfish or cooperative.

We define a node's trust level as a continuous real number in the range [0, 1], with 1 indicating complete trust, 0.5 ignorance, 0 complete distrust. The overall trust value is derived based on four components reflecting a node's status in terms of energy level (degree of remaining energy $\leq$ energy threshold, T_{energy}), honesty (i.e., degree of being uncompromised), cooperation (i.e., degree of being cooperative or unselfish), and closeness (i.e., number of 1-hop neighbors). In order to calculate trust values probabilistically, we develop analytical models based on SPN techniques to obtain each trust component. The obtained trust component values are used as the basis of deriving trust values probabilistically using our proposed trust metric below. Note that in practice energy and honesty trust component values can be binary as 0 or 1 while the cooperation trust component value can be a probability based on statistical data available for packet dropping behaviors during a trust update interval. Closeness is expressed by the number of 1-hop neighbors to represent relative largeness of 1-hop neighbors.

Now we address how the trust value toward a node is calculated. Our trust metric reflects four trust components as mentioned above: cooperation, energy, closeness, and honesty. The trust value ($T_{i,j}^{n-hop}(t)$) of node j as evaluated by node i using a trust chain with n hops is given by

$$\begin{aligned} T_{i,j}^{n-hop}(t) = {} & w_1 T_{i,j}^{n-hop,cooperation}(t) + w_2 T_{i,j}^{n-hop,energy}(t) \\ & + w_3 T_{i,j}^{n-hop,closeness}(t) + w_4 T_{i,j}^{n-hop,honesty}(t) \end{aligned} \tag{1}$$

The four trust components shown in Eq. (1) are weighted equally with $w_1 = w_2 = w_3 = w_4$ where $w_1 + w_2 + w_3 + w_4 = 1$ in our study. Now we explain how each trust component, $T_{i,j}^{n-hop,X}(t)$, where X=cooperation, energy, closeness, or honesty, is computed.

Given n hops as the maximum length of a trust chain for node i to find trust information about node j, node i can update node j's trust value at time t with both direct (or self) and indirect information. If node j cannot be found within n hops, node i relies on node j's past trust value with some decay factor. Reflecting these two cases $T_{i,j}^{n-hop,X}(t)$ is given by

$$\begin{aligned} T_{i,j}^{n-hop,\,X}(t) \quad & = \alpha(\beta T_{i,j}^{n-hop,\,X}(t-\Delta t) + (1-\beta) T_{i,j}^{n-hop,\,indirect-X}(t)) \\ & \quad + (1-\alpha) T_{i,j}^{n-hop,\,X}(t-\Delta t)' \end{aligned} \tag{2}$$

$$T_{i,j}^{n-hop,\,X}(t)' = e^{-\rho} T_{i,j}^{n-hop,\,X}(t) \quad \text{where} \quad \rho = \Delta t(\mathrm{h}) \tag{3}$$

The first term with probability α is for the case in which node j is found within n hops from node i's location, and both direct and indirect information is used to derive the trust value of node j evaluated by node i. The second term with probability $(1-\alpha)$ is for the case in which node j is not found within n hops from node i's location and the trust value at time t is evaluated based on past trust information at time $(t-\Delta t)$ with the decay factor to consider the staleness (ρ), as shown in Eq. (3). Note that Eqs. (2) and (3) are applied only when node i exists in the system. When node i does not exist in the system due to energy depletion or eviction by IDS, node i's trust value will drop to zero. In Eq. (2), β is used as a weight parameter for the node's own information, say "self-information" based on the past experience using trust value at time $(t-\Delta t)$ and $(1-\beta)$ is a weight parameter for indirect information using recommendations, say "others-information." In Eq. (2), α is the so called *trust availability* mentioned earlier, and can be obtained probabilistically from our SPN model by

$$\alpha = \frac{P_{i,j}^{n-hop}(t)}{P_{i,j}^{max-hop}(t)} \tag{4}$$

where *max* indicates the maximum *possible* hops that node i can use to find node j while n refers to the maximum *allowed* hops for node i to use for finding node j in the operational area. We obtain α probabilistically from our SPN model but in practice, $P_{i,j}^{n-hop}(t)$ is 1 when node j is found within n hops from node i's location, otherwise, it is 0. Further, $P_{i,j}^{max-hop}(t)$ is1 when node j is still alive in the system (that is, not compromised, or energy depleted); otherwise, it is 0. The probability that nodes i and j are within n hops, $P_{i,j}^{n-hop}(t)$, shown in Eq. (4), is obtained via

$$P_{i,j}^{n-hop}(t) = \sum_{k=1}^{n} q_{i,j}^{k-hop}$$

where

$$q_{i,j}^{k-hop}(t) = \sum_{(l,m)\epsilon S} (P_i^{loc=l}(t) P_j^{loc=m}(t)) \tag{5}$$

Here S is a set covering all (l, m) pairs with the distance between areas l and m being k hops. Notice that $P_{i,j}^{n-hop}(t)$ is the probability that hop distance between two nodes $\leq n$ hops, while $q_{i,j}^{k-hop}(t)$ is the probability that the hop distance between two nodes exactly equals k. $P_i^{loc=k}(t)$ is the probability that live node i is located in area k and can be directly obtained from our SPN model for statistical calculation. As mentioned above, in practice, $P_{i,j}^{n-hop}(t)$ has binary value (0 or 1), and Eq. (5) is used to obtain it probabilistically from our SPN model.

In Eq. (2), the indirect information for trust component X is computed by

$$T_{i,j}^{n-hop,\,indirect-X}(t) = \sum_{k=1}^{n} \left\{ \left(\frac{\sum_{m \in S} (T_{i,m}^{k-hop,PR}(t) T_{m,j}^{direct-X}(t))}{|S|} \right) \left(\frac{q_{i,j}^{k-hop}}{P_{i,j}^{n-hop}(t)} \right) \right\} \tag{6}$$

$$T_{i,m}^{k-hop,PR}(t) = T_{i,m1}^{PR}(t) T_{m1m2}^{PR}(t) \ldots T_{m(k-2),m}^{PR}(t) \quad \text{where} \quad k > 2 \tag{7}$$

$$T^{1-hop,PR}_{i,m}(t)=T^{PR}_{i,m}(t),\quad T^{2-hop,PR}_{i,m}(t)=T^{PR}_{i,m}(t) \tag{8}$$

where m is a functional recommender

$$T^{PR}_{i,w}=T^{direct\text{-}honesty}_{i,w}(t)T^{direct\text{-}cooperation}_{i,w}(t) \tag{9}$$

In Eq. (6), S is the set of functional recommenders having the highest trust values toward node j. In Eq. (6), $T^{k-hop,PR}_{i,m}(t)$ refers to the path reliability for the k-hop route between nodes i and m. $T^{k-hop,PR}_{i,m}(t)$ is specified in Eq. (7) for $k>2$. As shown in Eq. (8), when $k\leq 2$, m is the functional recommender of node j as well as the 1-hop neighbor of node i. This process reflects the *incomplete transitivity* characteristic of trust. Note that based on Josang and Pope (2005), we use referral trust with $T^{k-hop,PR}_{i,m}(t)$ and functional trust with $T^{direct-X}_{m,j}(t)$. Further, $T^{n-hop,\ indirect-X}_{i,j}(t)$ is derived based on direct trust relationships between all intermediate nodes from node i to node j in order to avoid the incorrect trust derivation problem indicated in Josang and Pope (2005). Eq. (9) describes how the path reliability between nodes i and w is obtained based on direct observations of node w's honesty and cooperation by node i.

The trust value of node j based on direct observations by node i, for a trust component X at time t, $T^{direct-X}_{i,j}(t)$, is obtained by

$$T^{direct-X}_{i,j}(t)=min\left[\frac{P^{X}_{j}(t)}{P^{X}_{i}(t)},\quad 1\right] \tag{10}$$

$$T^{direct-X}_{i,j}(0)=0.5 \tag{11}$$

In Eq. (10), we reflect the *subjective* characteristic of trust by dividing node j's capability ($P^{X}_{j}(t)$) by node i's capability ($P^{X}_{i}(t)$) where $P^{X}_{i}(t)$ for all j's is obtained directly from our analytical model based on SPN techniques. As Eq. (11) shows, we also assume that all nodes are trustable having ignorance trust value 0.5 in the beginning, say at time $t=0$.

$P^{X}_{i}(t)$ values for X=cooperation, energy, or honesty are directly obtained from our SPN while the one for X=closeness should be computed based on location information ($P^{loc\,=\,l}_{i}(t)$ where l indicates a particular area) obtained from our SPN model. $P^{closeness}_{i}(t)$ refers to the degree of node i's closeness to any neighboring node j at time t and is computed by

$$P^{closeness}_{i}(t)=\frac{\sum_{j\,=\,1,\ j\,\neq\,i}^{N}P^{1-hop}_{i,j}(t)}{\sum_{j\,=\,1,\ j\,\neq\,i}^{N}P^{max-hop}_{i,j}(t)} \tag{12}$$

$\sum_{j\,=\,1,\ j\,\neq\,i}^{N}P^{1-hop}_{i,j}(t)$ indicates the number of 1-hop neighbors of node i at time t and $\sum_{j\,=\,1,\ j\,\neq\,i}^{N}P^{max\text{-}hop}_{i,j}(t)$ is the number of all live nodes in the system except node i at time t. Thus, $P^{closeness}_{i}(t)$ is the average closeness of node i toward any node j.

We also derive the values of objective trust on each node in order to compare it against the trust value calculated by each node, the so called *subjective trust*, based on Eqs. (1)–(12). The objective trust is calculated based on the actual information of each node without considering trust decay over time and the length of a trust chain. The objective trust of node i is calculated by

$$T^{obj}_{j}(t)=w_1P^{cooperation}_{j}(t)+w_2P^{energy}_{j}(t)+w_3P^{closeness}_{j}(t)+w_4P^{honesty\prime}_{j}(t) \tag{13}$$

As Eq. (1), the four trust components shown in Eq. (1) are weighted equally with $w_1=w_2=w_3=w_4$ where $w_1+w_2+w_3+w_4=1$ in our study. Here $P^{X}_{j}(t)$ values for X=cooperation, energy, closeness, or honesty′ are component of the overall trust value. Note that $P^{honesty\prime}_{j}(t)$ is used for computing objective trust in Eq. (13) while $P^{honesty}_{j}(t)$ is used for deriving the measured subjective trust using Eqs. (1)–(12). In $P^{honesty\prime}_{j}(t)$, undetected compromised nodes are also considered as dishonesty. The difference between $P^{honesty}_{j}(t)$ and $P^{honesty\prime}_{j}(t)$) is that $P^{honesty\prime}_{j}(t)$ reflects actual compromised nodes, including compromised nodes that were not detected by IDS while $P^{honesty}_{j}(t)$ only accounts for the compromised nodes detected by IDS. Note that we do not assume that the IDS is perfect (i.e., non-zero false positives and false negatives). The objective trust in Eq. (13) is devised to refer to *trustworthiness* mentioned in Josang and LoPresti (2004) and Solhaug et al. (2007), meaning the objective level of trust probability.

The goal of this study is to identify an optimal trust chain length that can generate the measured subjective trust values (based on Eqs. (1)–(12)) most accurately but safely (not revealing risk) when compared with the objective trust values based on Eq. (13).

4.3. Energy model

We associate the energy level of a node with its state: compromised or not, selfish or not, and member or not. Depending on the amount of remaining energy, each node acts differently. The rate of energy consumption is also affected by the node's status. Thus, these parameters are interwoven and affect the node's lifetime significantly.

The proposed GCS must handle events that trigger various types of messages, including beacon, group communication, rekeying, status exchange, and intrusion detection (IDS) messages, as well as bogus messages from possible attackers. Recall that the status exchange message is used for trust evaluation of 1-hop neighboring nodes as well as other distant nodes. Each node transmits this status exchange containing the trust component values (for functional trust) of its 1-hop neighbors attached with its ID. Further, when the trust information for functional trust is forwarded, intermediate nodes will attach its direct referral trust about the previous node into the packet. A node will not forward a trust information packet when the received packet reaches the allowed length of the trust chain (in hops). More details of computing energy consumption are described in the Appendix of this paper.

4.4. Attack model

We consider the presence of outside and inside attackers by non-group members and legitimate group members, respectively. We assume that the existing prevention techniques such as encryption, authentication, or rekeying inhibit outsider attacks. Our trust management protocol will utilize IDS to detect inside attackers (compromised nodes) in order to achieve high survivability.

We assume that a node becomes compromised with a certain rate which can be obtained based on first-order approximation of historical attacker data in practice. We assume that an inside attacker, particularly compromised but not detected by IDS, may consume more energy in order to disseminate bogus messages, which is a way to perform a distributed denial of service (DDoS) attack. Further, inside attackers may modify or forge messages to disrupt normal group communication. We consider that selfish nodes in the system may perform packet dropping attacks. We also assume that smart inside attackers may not disrupt rekeying operations in order not to be easily detected by IDS. Further, they may want to have a chance to disrupt the entire system or fail a given mission by obtaining a secret key (i.e., group key) for performing more active attacks. A time-out mechanism embedded in performing a rekeying operation enables IDS to easily detect any node disrupting the rekeying operation. IDS will categorize nodes performing real attacks as "blacklisted" culprits and forward them to the GCS for permanent evictions of proven attackers through individual rekeying operations.

Please cite this article as: Cho J-H, et al. Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks. J Network Comput Appl (2011), doi:10.1016/j.jnca.2011.03.016

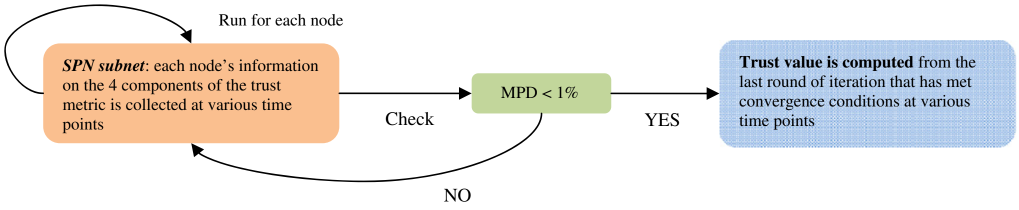
Run for each node
SPN subnet: each node's information on the 4 components of the trust metric is collected at various time points
Check
MPD < 1%
YES
NO
Trust value is computed from the last round of iteration that has met convergence conditions at various time points

Fig. 1. Hierarchical modeling processes using SPN subnets.

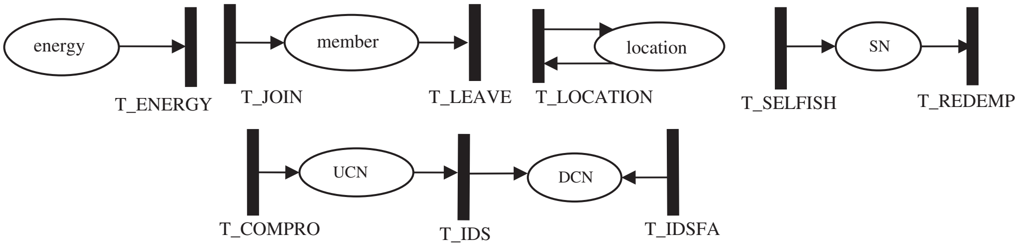

Fig. 2. SPN subnet model.

5. Performance model

5.1. Hierarchical modeling using SPNs

The goal of our study is to identify optimal design settings through the evaluation of mathematical models developed using a quantitative modeling technique. In this study, we use SPN as our modeling tool as it provides an efficient representation of a large number of states in the underlying *Markov* or *semi-Markov* model. We develop a hierarchical modeling technique based on SPN to avoid state explosion problems and to improve solution efficiency for realizing and describing a large-scale GCS. We use a SPN subnet to describe the behavior of a node in its lifetime in the presence of inside attackers (compromised nodes) and selfish nodes where IDS exists to detect the inside attackers. The square-shaped operational area consists of $m \times m$ sub-grid areas, each with width and height equal to the wireless radio range (R). Initially the location of each node is randomly distributed over the operational area based on uniform distribution. Nodes are assumed to be at the center of the sub-grids; thus each node has at most 4 neighboring areas. A node randomly moves to one of four locations in four directions (i.e., north, west, south, and east) in accordance with its mobility rate. The speed of each node is chosen from [0, 2) m/s based on uniform distribution at the beginning of network deployment, and then fixed during its lifetime. To avoid end-effects, movement is wrapped around (i.e., a torus is assumed).

The *SPN subnet* also gives us the location information of each node such as the probability that a node is in a particular location at time t, for example, the probability that node i is located in area j at time t. This information along with the information of other nodes' location information at time t provides the information to a node about its n-hop neighbors at time t. Since node movements are independent, the probability that two nodes are in a particular location at time t, is given by the product of the two individual probabilities. Further, the *SPN subnet* is used to obtain each node's information (i.e., degree of cooperation, energy, closeness, and honesty) to derive the trust relationships with other nodes in the system. An iterative technique is used for one SPN subnet to obtain other nodes' information from their SPN subnets since one subnet only describes one node's lifetime. In the first round of iteration, since there is no information available about the 1-hop neighbors, each area is assumed to have an equal number of nodes and they all are assumed to be healthy, meaning unselfish and uncompromised. In the second round of iteration, based on the information collected (i.e., number of healthy, selfish, or undetected compromised nodes as 1-hop neighbors) and location information (i.e., probability that each node is located in a particular area at time t) from the previous round of iteration, each node knows how many nodes are 1-hop neighbors that can directly communicate with it and their conditions whether they are selfish or compromised, and further how many 1-hop neighbors it has at time t. It then adjusts its conditions of 1-hop neighbors at time t with the outputs obtained from the jth round of iteration as inputs to the $(j+1)$th round of iteration. This process continues until a specified convergence condition is met. The Mean Percentage Difference (MPD) is used to measure the difference between critical design parameter values, a node's energy level, selfish probability, and undetected compromised probability, at time t in two consecutive iterations. Note that MPD is used to guarantee solution accuracy of the model output such as the actual trust information about each node in the network while utilizing the hierarchical SPN modeling technique. It is unrelated to the performance of the proposed trust protocol. The iteration stops when MPD is below a threshold (1%) for all nodes in the system. The estimated MPD with parameter X for node i is computed by

$$MPD_i^X = \frac{\sum_t^{max} D_i^x(t)}{N_{interval}} \quad \text{where} \quad D_i^x(t) = \frac{\left|X_i^{j+1}(t) - X_i^j(t)\right|}{X_i^j(t)} \tag{14}$$

where $X_i^j(t)$ indicates the value of parameter X of node i at time t in the jth round of iteration, max is the maximum time, and $N_{interval}$ is the number of time points. We compute MPD for the energy level, probability of being selfish, and probability of being compromised but not detected. The SPN subnet after convergence provides the actual node information with which we can calculate $P_i^X(t)$ for X=energy, cooperation or honesty in Eq. (10) and $P_i^{closeness}(t)$ in Eq. (12). Fig. 1 gives the brief overview of technical procedures performed using our SPN model.

Fig. 2 shows the *SPN subnet* model. The subnet describes a node's mobility behavior, whether the node is a member or not, and whether a node is compromised/detected by IDS or not, and whether a node is selfish or not. The SPN subnet gives the probability of each node being located in a particular area at a particular time point. The transition T_LOCATION is triggered when a node moves to a randomly selected area out of four different directions from its current location with the rate

Please cite this article as: Cho J-H, et al. Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks. J Network Comput Appl (2011), doi:10.1016/j.jnca.2011.03.016

calculated as S_{init}/R based on initial speed (S_{init}) and wireless radio range (R).

Depending on the randomly selected location, the number of tokens in place *location* is adjusted. We assume that inter-arrival times of a node's join and leave requests are exponentially distributed with rates λ and μ, respectively.

Place *energy* represents the current energy level of a node. An initial energy level is assigned to each node: we randomly generate a number in the range of [12, 24] hrs based on uniform distribution. A token is taken out when transition T_ENERGY fires. The transition rate of T_ENERGY is adjusted on the fly based on a node's state; it is lower when a node becomes selfish to save energy or when a node changes its status from member to non-member; it is higher when the node becomes compromised since it could perform attacks and consume more energy. We assume that T seconds will be taken to consume one energy token when a member node has no selfish or compromised 1-hop neighbors.

We use the energy model described in the Appendix for adjusting the time taken to consume one token in place *energy* based on a node's status. Specifically, we derive P for the energy consumption per second as a healthy node (Eq. (25)), $P_{selfish}$ for the energy consumption per second as a selfish node (Eq. (27)), $P_{non\text{-}member}$ for the energy consumption per second as a non-member (Eq. (28)), and $P_{attacker}$ for the energy consumption per second as an undetected attacker (Eq. (29)). Consequently, when a node is a healthy member, a token is consumed in $T(=(P\times T)/P)$ seconds; when a node is a selfish member, it takes $(P\times T)/P_{selfish}$ seconds; when a node is an undetected compromised member, $(P\times T)/P_{attacker}$ is taken; and when a node is non-member, $(P\times T)/P_{non\text{-}member}$ is taken. Therefore, depending on the node's status, its energy consumption is dynamically changed and accordingly its behaviors are affected. Place *UCN* indicates an undetected compromised node. Place *DCN* represents a detected compromised node. A node is compromised when transition T_COMPRO with rate λ_{com} fires where λ_{com} is the base compromising rate initially given. In practice, λ_{com} can be derived via first-order approximation from the observations of historical attack behaviors. The behavior of a node being compromised is associated with the energy level of the node. If the node has low energy, it is more likely to become compromised, and vice versa. This is modeled by the enabling function of T_COMPRO, which returns 1 to enable T_COMPRO or returns 0 to disable T_COMPRO, as follows:

$$\begin{aligned}&\text{enabling\_T\_COMPRO}:\\ &\quad \textit{if}\ (\textit{mark}(\textit{energy})>0\ \&\&\ \text{mark}(\text{UCN})==0\ \&\&\ \text{mark}(\text{DCN})\\ &\qquad ==0\ \&\&\ \text{mark}(\text{member})>0)\\ &\qquad \{\textit{if}(N_{rand}\le P_{dishonest})\ \textit{return}\ 1;\ \textit{else return}\ 0;\}\\ &\qquad \text{where}\quad N_{rand}=\textit{rand}[0,1](\textit{mark}(\textit{energy})+1)\end{aligned}\tag{15}$$

Here rand [0, 1] returns a random real number in the range of [0, 1] based on uniform distribution and *mark*(*energy*) indicates the remaining energy. $P_{dishonest}$ models the inherent behavioral nature of a node's dishonesty and is a randomly selected number based on the truncated exponential distribution with mean 0.5 and range of [0, 1]. Eq. (15) implies that a node behaves dishonestly based on the random seed of the bad behavior as its nature but the bad behavior can be relaxed or further enhanced based on the current energy level of the node. If the node is compromised, a token goes to *UCN*, being compromised but not being detected by IDS. While the node is not detected by IDS, it has a chance to perform attacks. After it is being detected by IDS, a token is taken out from *UCN* into *DCN* and the node is evicted immediately through individual rekeying operations.

We consider false alarm probabilities of IDS. False negative probability (P_{fn}^{IDS}) of IDS is applied in T_IDS with the rate $(1-P_{fn}^{IDS})/T_{IDS}$ and false positive probability (P_{fp}^{IDS}) of IDS is considered in T_IDSFA with the rate P_{fp}^{IDS}/T_{IDS}.

Place *SN* represents whether a node is selfish or not. If a node becomes selfish, a token goes to *SN* by triggering T_SELFISH. Transition T_SELFISH fires based on the condition of the energy level of a node. Our assumption is that if the node has low energy, it is more likely to become selfish, and vice versa. The enabling functions for T_SELFISH and T_REDEMP are given in Eqs. (16) and (17), respectively by

$$\begin{aligned}&\text{enabling\_T\_SELFISH}:\ \textit{if}\ (\textit{mark}\ (\textit{energy})>0\ \&\&\ \text{mark (member)}\\ &\quad >0\ \&\&\ \text{mark (SN)}==0)\\ &\quad \{\textit{if}(N_{rand}\le P_{selfish})\ \textit{return}\ 1;\quad \textit{else return}\ 0;\}\\ &\quad \text{where}\quad N_{rand}=\textit{rand}[0,1](\textit{mark}(\textit{energy})+1)\end{aligned}\tag{16}$$

$$\begin{aligned}&\text{enabling\_T\_REDEMP}:\ \textit{if}(\textit{mark}(\textit{energy})>0\ \&\&\ \text{mark(member)}\\ &\quad >0\ \&\&\ \text{mark(SN)}>0)\\ &\quad \{\textit{if}(N_{rand}\le P_{selfish})\ \textit{return}\ 0;\quad \textit{else return}\ 1;\}\\ &\quad \text{where}\quad N_{rand}=\textit{rand}[0,1](\textit{mark}(\textit{energy})+1)\end{aligned}\tag{17}$$

$P_{selfish}$ models the inherent behavioral nature of a node's selfishness and is a randomly selected number based on the truncated exponential distribution with mean 0.5 and range of [0, 1]. Other parameters are similar to those used in Eq. (15). We define T_{gc} as the time interval to disseminate a group communication packet, assumed to be exponentially distributed in this work. Each node's selfishness is checked whenever a group communication packet is transmitted, so that the transition rate of T_SELFISH is $1/T_{gc}$. The transition T_SELFISH is triggered when a node is a member, alive with remaining energy ($\textit{mark}(\textit{energy})>0$), and currently not selfish. When the randomly selected number reflecting the degree of the node's current energy level (N_{rand}) is less than the probability of being selfish ($P_{selfish}$), T_SELFISH fires, and vice versa. We also similarly model the redemption mechanism for selfish nodes by using the transition T_REDEMP with rate $1/T_{trust}^{update}$. A node can have a chance to be redeemed in a reevaluation period, corresponding to a trust update interval (T_{trust}^{update}). During the reevaluation period, if the node behaves well, redemption is awarded, and vice versa. If a node has sufficiently low energy, it may choose to remain selfish to save its energy in a similar way as in transition T_SELFISH. No redemption service is provided for compromised nodes, whether they are detected or not.

5.2. Calculation of trust components

The trust value of node j by node i is calculated based on the actual information (i.e., $P_i^X(t)$ for energy, cooperation and honesty and $P_i^{loc=k}(t)$ to derive $P_i^{closeness}(t)$) on nodes collected from the last round of iteration from the SPN subnet that has met the convergence condition. We obtain the four trust component values based on a reward assignment technique described below.

For each trust component calculation from the SPN subnet of a particular node, with $X(t)$ representing a general property value at time t, the reward function would be

$$X(t)=\sum_{i\in S}(r_i\times P_i(t))\tag{18}$$

Here S indicates a set of states that meet particular conditions, $P_i(t)$ is the probability that the system is in state i at time t, and r_i is the reward to be assigned to those states.

Table 1 lists the conditions to be satisfied (in the "if" part) and the reward r_i, used (in the "return" part) in each reward function. Note that for $P^{honesty}(t)$, we reflect past experiences of a node's healthiness in order to consider the case when a node may detect that a target is compromised, even if it is not having been detected by IDS. We also differentiate $P^{honesty}(t)$ used for the subjective trust

Table 1
Reward functions.

Component	Reward returned based on conditions in S

$P^{energy}(t)$	$if(mark(energy) > T_{energy}$ & mark(DCN) $= = 0$
	& mark(member) $> 0)$ return 1;
	otherwise return 0;

$P^{cooperation}(t)$	$if(mark(SN) = = 0$ & mark(member) > 0 & $mark(energy) > 0$ & mark(DCN) $= = 0)$ *return*1; *othewise return* 0;
$P^{honesty}(t)$	$if(mark(DCN) = = 0$ & mark(UCN) $= = 0$ & $mark(member) > 0$ & mark(energy) $> 0)$ return1; $elseif(mark(DCN) = = 0$ & mark(UCN) > 0 & $mark(member) > 0$ & mark(energy) $> 0)$ $returnP^{healthy}(t - \Delta t)$; *otherwise return* 0;
$P^{honesty'}(t)$	$if(mark(DCN) = = 0$ & mark(UCN) $= = 0$ & $mark(member) > 0$ & mark(energy) $> 0)$ return1; *otherwise return* 0;
$P^{loc=k}(t)$	$if(mark(location) = = k$ & mark(member) > 0 & mark(energy) > 0 & mark(DCN) $= = 0)$ *return*1; *otherwise return* 0;
$P_{UCN}^{loc = k}(t)$	$if((mark(member) > 0)$ & (mark(UCN) $> 0)$ & (mark(energy)0) & (mark(location) $= =$ k)) *return*1; *otherwise return* 0;
$P_{selfish}^{loc = k}(t)$	$if(mark(member) > 0$ & mark(SN) > 0 &mark(energy) > 0 & mark(location) $= = k$ & mark(DCN) $= = 0)$ *return*1; *otherwise return* 0;
$P_{healthy}^{loc = k}(t)$	$if(mark(member) > 0$ & mark(SN) $= = 0$ & $mark(energy) > 0$ & mark(DCN) $= 0$ & $mark(location) = = k)$ *return*1; *otherwise return*0;

Table 2
Default parameter values used.

Param	Value	Param	Value	Param	Value
N	150	S_{init}	(0, 2] m/s	M_{beacon}	32 bits
$\lvert S\rvert$	3	β	0.8	M_{rekey}	128 bits
R	250 m	T_{beacon}	2 min	M_{status}	800 bits
λ	1/(1 h)	T_{status}	10 min	M_{bogus}	300 bits
μ	1/(4 h)	λ_{com}	1/(4 h)	M_{gc}	500 bits
T_{IDS}	10 min	T_{trust}^{update}	10 min	M_{IDS}	128 bits
P_{fn}^{IDS}	0.5%	E_{init}	[12, 24] h	T_{bogus}	0.5 min
P_{fp}^{IDS}	0.5%	T_{gc}	2 min	T	1 hr

from $P^{honesty'}(t)$ employed for the objective trust in that the latter also considers the dishonesty of a node that is not detected by IDS but is compromised at time t, as shown in Table 1. Based on $P_j^{loc = k}(t)$ obtained, the probability that two nodes are k hops away can be predicted and used to calculate $P_j^{closeness}$. Note that in Table 1, we omit the symbol j for notation efficiency.

6. Numerical results and analysis

In this section, we show numerical results obtained from evaluating our hierarchical SPN model. Table 2 gives the default parameter values used in this study. The square-shaped operational area consists of 6×6 sub-grid areas, each with width and height equal to the wireless radio range $(R) = 250$ m.

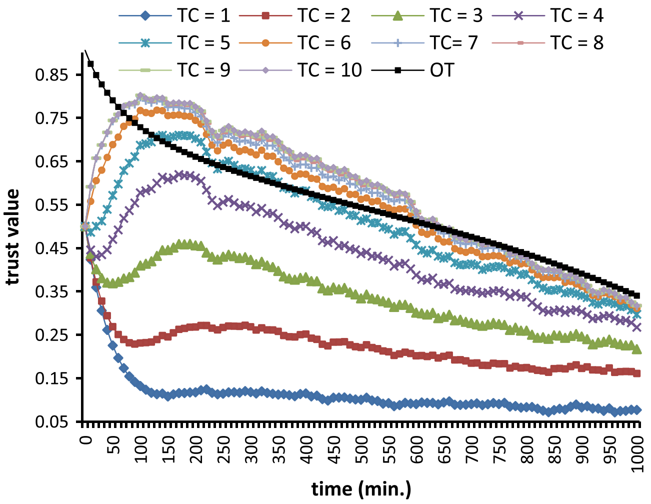

Fig. 3. Trust values over time with respect to the length of a trust chain—one target node's trust values evaluated by one evaluator node.

Fig. 3 describes how the trust value of one node evaluated by one evaluator node evolves over time for various values of the trust chain (TC) length. As explained earlier, based on Josang and LoPresti (2004) and Solhaug et al. (2007), when the measured subjective trust overestimates the objective trust (called *trustworthiness* in Josang and Solhaug's terminology), collaboration based on the measured trust may be risky. Thus, considering the hostile environment of battlefield situations, our goal is to identify the optimal TC length that gives subjective trust values closest to the objective trust based on Eq. (13).

We consider two baseline cases against which our proposed trust protocol using the optimal TC length is compared. The first baseline case ("local trust evaluation") is that trust of a trustee is evaluated only based on local information collected by direct observations or information from 1-hop neighbors of a trustor. This case corresponds to the trust evaluation with TC length = 1. The second baseline case ("global trust evaluation") is that trust of the trustee is assessed based on trust evidences forwarded by nodes that have direct evidences towards the trustee using the maximum possible length of the trust chain (TC length = 10 in this case study).

When the TC length is short, we observe low trust values because of the low chances of finding the target node (low trust availability) even if the path reliability is high. When the TC length becomes longer, higher trust values are observed due to higher chances of finding the target node (high trust availability) even if the path reliability is low. Thus, the effect of trust availability exceeds that of path reliability. Notice that when TC is sufficiently long, say TC length > 5, trustworthiness is overestimated, which should be avoided to reduce risk. Thus, each node selects a TC length that provides accurate but safe trust values by considering the desired trust availability and path reliability over time. In the scenario depicted in Fig. 3, a TC length of 4 appears to be optimal over the entire time horizon. Notice that local trust evaluation with TC length = 1 shows very low trust values with high inaccuracy due to lack of information. On the other hand, global trust evaluation using TC length = 10 overestimates trust due to low path reliability, meaning that trust information is not delivered properly, and trust evaluation relies on past trust values which introduce inaccuracy.

Now, we explain the main underlying tradeoff addressed in our trust evaluation in terms of trust availability and path reliability. Fig. 4 shows how trust availability and path reliability change with TC length. Note that we use Eq. (4) to obtain the normalized trust availability and Eqs. (7)–(9) to compute path reliability. Intuitively, path reliability decreases as TC length

Please cite this article as: Cho J-H, et al. Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks. J Network Comput Appl (2011), doi:10.1016/j.jnca.2011.03.016

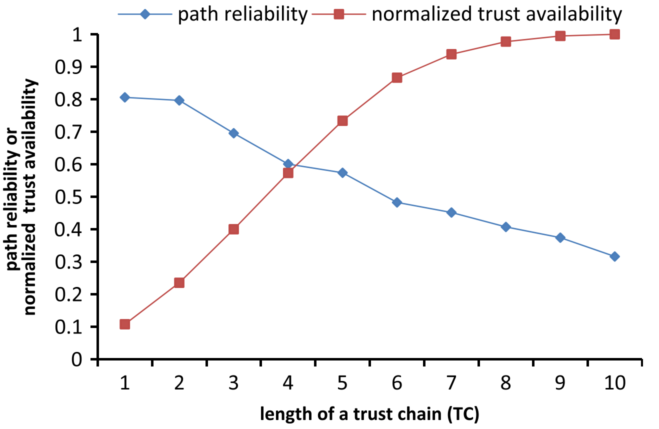

Fig. 4. Path reliability versus normalized trust availability in the process of one node's trust evaluation for one target node.

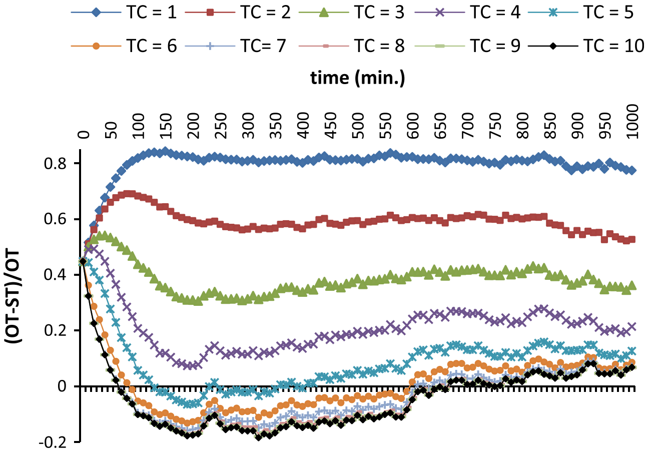

Fig. 5. Accuracy of subjective trust when comparing with objective trust with respect to the length of a trust chain—one target node's trust accuracy evaluated by one evaluator node.

grows since the functional trust sent from the initial functional recommender has to go through more intermediate nodes (referral recommenders) and this requires the cooperation and honesty of all the referral recommenders forwarding the reference. On the other hand, trust availability increases as TC length increases since allowing more hops to find a target node increases the possibility of finding a functional recommender. We observe that when TC=4, trust availability exceeds 57% while path reliability exceeds 60%. Based on this information, each node can adjust its TC length to obtain trust values achieving both desired trust availability and required path reliability.

Next we discuss the accuracy of the measured subjective trust by plotting the normalized difference between the measured subject trust (ST) and the objective trust (OT), (OT−ST)/ST as shown in Fig. 5. Recall that we would like this ratio to be as close to zero as possible, but positive (so as to avoid risk via overestimation). As in Fig. 4, we observe that all cases with TC length > 3 perform well, and that performance improves with increasing TC length as time increases. In particular, we observe that in the very beginning and end, TC=10 gives the most accurate trust values.

Fig. 6 shows the number of trustee nodes that have their most accurate trust values at a specific TC length as time progresses over time. Note that the trust values used to identify the best TC length is based on one node's evaluation. For example, we measure how many nodes have their most accurate trust values given a fixed TC length at various time points. The general trend observed is that in the first half of the observed time period, TC length > 3 is identified as the optimal TC length. In particular, it is noticeable that trust values are most accurately evaluated with TC length=10 in the very beginning since nodes do not have sufficient past trust information due to lack of interactions. But in the second half of the observation period, we also see that TC length < 4 generates the most accurate trust values. Notice that the overall best performance is observed with TC length=4 corresponding to the crossing point TC length=4 of path reliability and trust availability in Fig. 4.

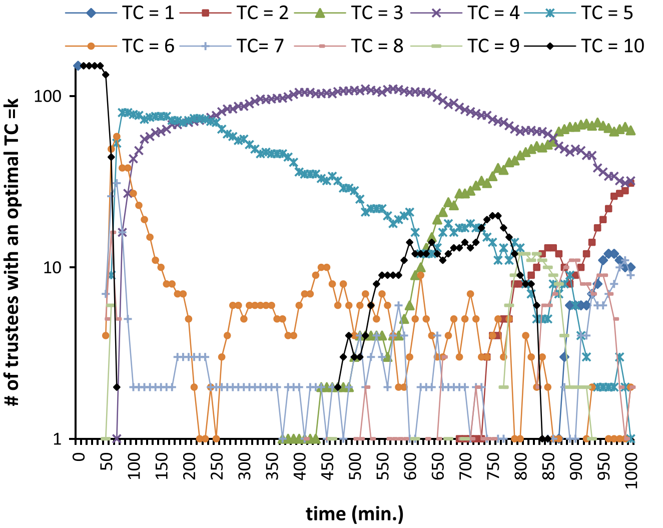

Fig. 6. Number of trustee nodes where using an optimal TC=k shows the most accurate trust values without revealing risk-all nodes evaluated by one evaluator node.

7. Conclusions and future work

We developed and analyzed a trust management protocol for a mission-driven GCS in MANETs for efficiently establishing accurate trust relationships among participating nodes that have not had any prior interactions. We defined a composite trust metric that takes into account both QoS and social trust aspects. We utilized a hierarchical SPN model to describe the behaviors of a node to tradeoff trust availability for path reliability over space. The model allows us to identify the optimal length of a trust chain to accurately evaluate the trust level of participating nodes on the trust chain. Our trust metric reflects subjectivity, asymmetry, dynamicity, incomplete transitivity, and context-dependency, which are key aspects of the trust concept in MANETs. Our conclusion is that an optimal trust chain length exists such that it can most accurately evaluate trust values of participating nodes without overestimation. Each node can adopt an optimal trust chain length that meets desired trust availability and path reliability and use the evaluated trust values for collaboration decision with other nodes in the network.

As our ongoing work (Cho et al., 2010), we will further investigate the complexity issue of the tradeoff between trust accuracy and resource consumption (i.e., communication overhead) in MANETs as the length of the trust chain increases.

We plan to extend our research by (1) identifying optimal design settings given a mission with various difficulty levels or requirements; (2) examining tradeoffs between system lifetime (based on a required trust level) and performance (e.g., service response time); (3) applying more realistic group mobility models; (4) validating the proposed trust protocol via simulations in a dynamic network scenario; and (5) comparing the proposed trust protocol with existing protocols to evaluate the performance of applications such as intrusion detection, secure routing, and key management.

Please cite this article as: Cho J-H, et al. Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks. J Network Comput Appl (2011), doi:10.1016/j.jnca.2011.03.016

Appendix

The approximate energy consumption per bit at a transmitter (Kansal et al., 2005) is

$$P_t(i,j) = \alpha_1 d(i,j)^v \tag{19}$$

where $d(i,j)$ is the distance between transmitter i and receiver j, v is the path-loss factor (typically, $2 \le v \le 6$), α_1 is a distance independent parameter. For simplicity, we use $\alpha_1 = 10^{-11}$ (Kansal et al., 2005), and $d(i, j) = R$ (wireless radio range). Since the nodes are assumed to be located at the center of each area, we have $P_t = 10^{-11}(R)^2$ for the energy consumption per bit at a transmitter, assuming $v = 2$. The energy consumed per second in data transmission by a node is given by

$$P_{send} = P_t[A + BN^{healthy}_{1-hop} + CN^{selfish}_{1-hop} + DN^{UCN}_{1-hop}] \tag{20}$$

Here the first term indicates energy consumption for its own transmission where A represents bits to be transmitted per second, considering messages for beacon, group communication, IDS-related activities, status exchange, and rekeying operations. The second term represents energy consumption for forwarding packets for healthy 1-hop neighbors ($N^{healthy}_{1\text{-}hop}$), that are neither compromised nor selfish, where B represents bits to be transmitted per second, reflecting messages for group communication, IDS-related activities, and rekeying operation. Beacon and status exchange messages are not required to be disseminated to all group members, and so are excluded from forwarding. The third term indicates the energy consumption for transmitting packets from selfish 1-hop neighbors ($N^{healthy}_{1\text{-}hop}$) that do not forward group communication packets received from others where C indicates bits to be transmitted per second for IDS-related activities, rekeying related operations, and its own group communication. The fourth term represents the packet transmission from compromised but undetected 1-hop neighbors ($N^{UCN}_{1\text{-}hop}$) where D is the same as A plus bits for transmitted bogus messages. Our analytical model is a prediction model that reflects possible traffic generated by potential attackers. $N^{healthy}_{1\text{-}hop}$ is set to the average 1-hop neighboring nodes (neither selfish nor compromised), $N^{selfish}_{1\text{-}hop}$, and $N^{UCN}_{1\text{-}hop}$ are set to zero in the first round of iteration of the SPN subnets based on the assumption that all neighbors are healthy. From the second round of iteration, the estimations of $N^{healthy}_{1\text{-}hop}$, $N^{selfish}_{1\text{-}hop}$, and $N^{UCN}_{1\text{-}hop}$ obtained at the end of the previous round of iteration are used. Note that $N^{healthy}_{1-hop}$, $N^{selfish}_{1-hop}$, and N^{UCN}_{1-hop} are time-averaged values; they reflect the average behaviors of the system and can be estimated after the first round of iteration where the global location information is available.

Node i's $N^{healthy}_{i,1-hop}$ is calculated as

$$N^{healthy}_{i,1-hop} = \frac{\sum_{t=0}^{max} \sum_{k=1}^{N_{area}} P_i^{loc=k}(t)[\sum_{x \in X} L^{loc=x}_{i,healthy}(t)]}{N_{interval}} \tag{21}$$

$$L^{loc=x}_{i,healthy}(t) = \sum_{j \in S, i \notin S} P^{loc=x}_{j,healthy}(t) \tag{22}$$

In Eq. (21), max is the upper bound of time measured, N_{area} refers to the number of subareas in the operational area, and $N_{interval}$ is the number of time points used. $P_i^{loc=k}(t)$ is the probability that node is located in area k at time t. X is a set including 1-hop neighboring areas of node i and $L^{loc=x}_{i,healthy}(t)$ is the number of healthy (i.e., unselfish and uncompromised) nodes in area x except for node i.

In Eq. (22), S includes all nodes' ids except for node i and $P^{loc=x}_{j,healthy}(t)$ is the probability that node j is healthy and located in area x at time t. $P_i^{loc=k}(t)$ in Eq. (21) and $P^{loc=x}_{j,healthy}(t)$ in Eq. (22) are obtained from our performance model described in Section 5. Similarly, $N^{selfish}_{1-hop}$ and N^{UCN}_{1-hop} are calculated as Eqs. (21) and (22). To generalize terms, we omit the symbol i in $N^{healthy}_{i,1-hop}$, $N^{selfish}_{i,1-hop}$, and $N^{UCN}_{i,1-hop}$ in Eq. (20) and the following equations.

Assume that a node may leave the group voluntarily with rate μ and may rejoin the group with rate λ. Then, the probability that a node is in the group is $\lambda/(\lambda+\mu)$ and the probability that it is not is $\mu/(\lambda+\mu)$. T_{IDS} represents an interval for evictions of detected compromised nodes by IDS. Then, $T_{rekeying}$, a rekeying interval, is calculated as

$$T_{rekeying} = \frac{(1/\Lambda_{J+L} + T_{IDS})}{2} \quad \text{where} \quad \Lambda_{J+L} = \frac{2\lambda\mu}{\lambda+\mu} \tag{23}$$

where Λ_{J+L} is the aggregate join and leave rate. This indicates the join and leave rates of all current nodes in equilibrium.

When a packet is received, the energy consumed is half of transmission energy (Kansal et al., 2005) by using $P_r = P_t/2$ in this work. Each member node consumes energy per second for receiving packets from 1-hop neighbors as follows:

$$P_{receive} = P_r[AN^{healthy}_{1-hop} + EN^{selfish}_{1-hop} + DN^{UNC}_{1-hop}] \tag{24}$$

Note that A and D used here indicate the same number of bits used as in Eq. (20). E represents bits received per second for beacon, IDS-related, status exchange, rekeying, and its own group communication messages. Even though the system would not know undetected compromised nodes, this is a prediction model estimating the energy consumption with respect to receiving packets from potential attackers (i.e., undetected compromised nodes). The first term explains the energy consumption by receiving packets from healthy 1-hop neighbors ($N^{healthy}_{1-hop}$). The second term indicates the energy consumption by receiving packets from selfish 1-hop neighbors ($N^{selfish}_{1-hop}$). The third term represents the energy consumption by receiving packets from compromised but undetected 1-hop neighbors (N^{UNC}_{1-hop}).

In summary, the energy consumption per node per second is

$$P = P_{send} + P_{receive} \tag{25}$$

If a member node is selfish, it does not forward any packet from others. The energy consumption per second for data transmission by a selfish node is given by

$$P_{send,\ selfish} = P_t A \tag{26}$$

If a member node is selfish, energy consumption per second for receiving packets is also $P_{receive}$ since we assume all nodes are in promiscuous mode. Thus, the node will save $P_{send} - P_{send,selfish}$ energy by being selfish. Thus, the total energy consumption as a selfish node per second is

$$P_{selfish} = P_{send,\ selfish} + P_{receive} \tag{27}$$

If a node is a non-member, it will only transmit and receive beacons. The energy consumption per second as a non-member is computed as

$$P_{non\text{-}member} = P_{send,\ non\text{-}member} + P_{receive,\ non\text{-}member} = F(P_t + P_r N_{1-hop}) \tag{28}$$

Here N_{1-hop} includes $N^{healthy}_{1-hop}$, $N^{selfish}_{1-hop}$, and N^{UCN}_{1-hop} since any live nodes will disseminate beacon messages, and F indicates bits to be transmitted/received per second for a beacon message.

If a member node is compromised but not detected by IDS, the energy consumption must account for an additional bogus packet

(with G representing bits in a bogus message)

$$P_{attacker} = P_{send,\ attacker} + P_{receive} \quad \text{where} \quad P_{send,\ attacker} = P_{send} + P_t G \tag{29}$$

References

Abdul-Rahman A, Hailes S. Using recommendations for managing trust in distributed systems. In: Proceedings of IEEE Malaysia international conference on communication; 1997.

Adams WJ, Davis NJ. Toward a Decentralized Trust-based access control system for dynamic collaboration. In: Proceedings of sixth annual IEEE SMC information assurance workshop; 2005. p. 317–24.

Ahmed E, Samad K, Mahmood W. Cluster-based intrusion detection (CBID) architecture for mobile ad hoc networks. In: Proceedings of the AusCERT Asia pacific information technology security conference; 2006.

Albers P, Camp O, Percher J-M, Jouga B., Mé L., Puttini R. Security in ad hoc networks: a general intrusion detection architecture enhancing trust based approaches. In: Proceedings of first international workshop on wireless information systems; 2002, p. 1–12.

Balakirishnan V, Varadharajan V. Designing secure wireless mobile ad hoc networks. In: Proceedings of 19th international conference advanced information networking and applications, vol. 2; 2005. p. 5–8.

Baras JS, Jiang T. Cooperative games, phase transition on graphs and distributed trust in MANETs. In: Proceedings of 43th IEEE conference on decision and control, vol. 1; 2004. p. 93–8.

Bhargava B, Lilien L, Rosenthal A, Winslett M, Sloman M, Dillon TS, et al. The pudding of trust. IEEE Intelligent Systems 2004;19(5):74–88.

Blaze M., Feigenbaum J., Lacy J. Decentralized trust management. In: Proceedings of IEEE symposium on security and privacy; 1996. pp. 164–73.

Boukerche A, Ren Y. A security management scheme using a novel computational reputation model for wireless and mobile ad hoc networks. In: Proceedings of international workshop on modeling analysis and simulation of wireless and mobile systems; 2008. p. 88–95.

Capkun S, Buttyan L, Hubaux J–P. Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing 2003;2(1): 52–64.

Capra L. Toward a human trust model for mobile ad-hoc networks. In: Proceedings of second UK-UbiNet workshop; 2004.

Cho, JH, Chan, K, Swami, A, Rivera, B. On tradeoffs between trust accuracy and resource consumption in communications and social networks. In: Proceedings of the 27th army science conference; 2010.

Cho, JH, Swami, A, Chen, I.R. A survey of trust management in mobile ad hoc networks. IEEE Communications Surveys and Tutorials, online available, Dec. 2010.

Cho JH, Swami A, Chen IR. Modeling and analysis of trust management for cognitive mission-driven group communication systems in mobile ad hoc networks. In: Proceedings of the 2009 IEEE/IFIP international symposium on trusted computing and communications; 2009.

Ciardo G, Fricks RM, Muppala JK, Trivedi KS. SPNP users manual version 4, 6. Department Electrical Engineering, Duke University; 1994/1999.

Cook KS, editor. Trust in society, vol. 2. New York: Russell Sage Foundation Series on Trust; 2003.

Ghosh T, Pissinou N, Makki K. Towards designing a trust routing solution in mobile ad hoc networks. Mobile Networks and Applications 2005;10:985–95.

Golbeck J, editor. Computing with social trust. Human-Computer Interaction Series. Springer; 2009.

Gray E, O'Connell, Jensen C, Weber A, Seigneur J-M, Yong C. Towards a framework for assessing trust-based admission control in collaborative ad hoc applications. Technical Report, TCD-CS-2002-66, Trinity College Dublin; 2002.

Hadjichristofi GC, Adams WJ, Davis NJ. A framework for key management in a mobile ad hoc network. In: Proceedings of international conference on information technology: coding and computing, vol. 2; 2005. p. 568–73.

Jiang T, Baras JS. Ant-based adaptive trust evidence distribution in MANET. In: Proceedings of second international conference on mobile distributed computing systems workshops; 2004. p. 588–93.

Josang A, LoPresti S. Analyzing the relationship between risk and trust. In: Proceedings of second international conference trust management; 2004. p. 135–45.

Josang A, Pope S. Semantic constraints for trust transitivity. In: Proceedings of second Asia-Pacific conference on conceptual modeling; 2005.

Kansal A, Ramamoorthy A, Srivastava MB, Pottie GJ. On sensor network lifetime and data distortion. In: Proceedings of international symposium on information theory; 2005. p. 6–10.

Li H, Singhal M. Trust management in distributed systems. Computer 2007;40(2):45–53.

Li J, Li R, Kato J. Future trust management framework for mobile ad hoc networks: security in mobile ad hoc. IEEE Communications Magazine 2008;46(4):108–14.

Li R, Li J, Liu P, Chen HH. On demand public key management for mobile ad hoc networks. Wiley's Wireless Communications and Mobile Computing 2006;6(3):295–306.

Li L. Trust derivation and recommendation management in a trust model. In: Proceedings of the international conference on intelligent information hiding and multimedia signal processing; 2008. p. 219–22.

Luo H, Kong J, Zerfos P, Lu S, Zhang L. URSA: ubiquitous and robust access control for mobile ad hoc networks. IEEE/ACM Transactions on Networking 2004;12(6):1049–63.

Mahmoud Q, editor. Cognitive networks: towards self-aware networks. Wiley; 2007.

Marti S, Giuli T, Lai K, Baker M. Mitigating routing misbehavior in mobile ad hoc networks. In: Proceedings of sixth annual ACM/IEEE mobile computing and networking; 2000. p. 255–65.

Moe MEG, Helvik BE, Knapskog SJ. TSR: Trust-based secure MANET routing using HMMs. In: Proceedings of fourth ACM symposium on QoS and security for wireless and mobile networks; 2008. p. 83–90.

Moloney M, Weber S. A context-aware trust-based security system for ad hoc networks. In: Proceedings of first international conference on security and privacy for emerging areas in communication networks-workshop; 2005. p. 153–60.

Mundinger J, Boudec J. Analysis of a reputation system for mobile ad hoc networks with liars. Performance Evaluation 2008;65(3–4):212–26.

Nekkanti RK, Lee C. Trust based adaptive on demand ad hoc routing protocol. In: Proceedings of 42th annual ACM southeast regional conference; 2004. p. 88–93.

Pirzada AA, McDonald C. Establishing trust in pure ad hoc networks. In: Proceedings of 27th Australasian computer science conference, vol. 26; 2004. p. 47–54.

Plesse T, Lecomte J, Adjih C, Badel M, Jacquet P, Laouiti A, et al. OLSR performance measurement in a military mobile ad hoc network. In: Proceedings of 24th international conference on distributed computing systems; 2004. p. 704–9.

Sahner RA, Trivedi KS, Puliafito A. Performance and reliability analysis of computer systems. Kluwer Academic Publishers; 1996.

Solhaug B, Elgesem D, Stolen K. Why trust is not proportional to risk? In: Proceedings of second international conference on availability, reliability, and security; 2007. p. 11–8.

Soltanali S, Pirahesh S, Niksefat S, Sabaei M. An efficient scheme to motivate cooperation in mobile ad hoc networks. In: Proceedings of the international conference on networking and services; 2007. p. 98–103.

Steiner M, Tsudik G, Waidner M. Diffie-Hellman key distribution extended to group communication. In: Proceedings of third ACM conference on computer and communications security; 1996, p. 31–7.

Schoorman FD, Mayer RC, Davis JH. An integrative model of organizational trust: past, present, and future. Academy of Management Review 2007;31(2): 344–54.

Sun YL, Yu W, Han Z, Liu KJR. Information theoretic framework of trust modeling and evaluation for ad hoc networks. IEEE Journal on Selected Areas in Communications 2006;24(2):305–17.

Tardiff AJ, Gowens JW (editors). ARL advanced telecommunication and information distribution research program, Final Report; 2001.

Theodorakopoulos G, Baras JS. Trust evaluation in ad hoc networks. In: Proceedings of third ACM workshop on wireless security; 2004. p. 1–10.

Virendra M, Jadliwala M, Chandrasekaran M, Upadhyaya S. Quantifying trust in mobile ad-hoc networks. In: Proceedings of international conference integration of knowledge intensive multi-agent systems; 2005. p. 65–70.

Yan Z, Zhang P, Virtanen T. Trust evaluation based security solutions in ad hoc networks. In: Proceedings of seventh nordic workshop on security IT systems; 2003.

Yang L, Kizza JM, Cemerlic A, Liu F. Fine-grained reputation-based routing in wireless ad hoc networks. IEEE Intelligence and Security Informatics 2007:75–8.